IN THE CLAIMS

Please amend claims 1, 10, and 20 as indicated below.

1.   (Currently Amended) A method of compiling a source language routine, the method

comprising:

generating an intermediate language routine from the source language routine

provided by a user;

specifying an initial value of each routine variable by calling an initialization routine

associated withseparated from the source language routine and provided by the user, the

initialization routine being called from the intermediate language routine;

performing an optimizing change to the intermediate language routine that results in

an altered intermediate language routine, the optimizing change including a change based on a

criterion specified by the user;

generating a machine language routine from the altered intermediate language routine;

initializing the variables to the specified initial value;

executing the machine language routine in a computer system having architecture of a

target computer system using the initialized values;

measuring a characteristic of the execution by calling a measuring routine associated

with separated from the source language routine and provided by the user, the measuring

routine being called from the intermediate language routine; and

evaluating whether a stopping criterion after said executing is met based on a result of

calling the measuring routine and if not,

repeating said performing through said measuring, and

saving the machine language routine having a best measured characteristic,

until the stopping criterion is met, wherein said characteristic includes at least one of a

~~timing wherein the best measured timing is~~ a lowest timing, a machine language

routine size, and a bus utilization metric,

wherein the initialization routine and the measuring routine are provided by the user as

additional source code separated from the source language routine to be optimized, and

wherein the initialization routine and the measuring routine are compiled, linked, and

launched with the source language routine as a single application.

2. – 3. (Cancelled)

4.   (Original) The method defined in claim 1, further including defining a plurality of

segments within the intermediate language routine, each said segment comprising consecutive

intermediate language routine statements wherein no segment includes a same intermediate

language routine statement, and the performing an optimizing change is performed within one

of the segments.

5.   (Original) The method defined in claim 1 further including determining ordering

dependencies in said intermediate language routine wherein said performing an optimizing

change includes maintaining the determined ordering dependencies.

6.   (Original) The method defined in claim 1 wherein the optimizing change comprises

one of a generic optimization, a reordering, a user selectable reordering, a user selectable

global reordering, a user selectable insertion of at least one instruction in a selectable position

in the intermediate language routine, and a user selectable removal of at least one instruction from a selectable position in the intermediate language routine; wherein each optimizing change does not affect the intermediate language routine integrity.

7. (Original) The method defined in claim 1 further including after the generating the machine language routine and before the executing the machine language routine, at least one user selectable optimization to the machine language routine.

8. (Original) The method defined in claim 1 wherein the optimizing changes in a sequence of a plural number of a repeated said performing resulting from the stopping criterion not met is performed according to a process that includes at least one of a non-repeating optimizing change, a user selectable optimization change sequence; and a parallel search across a plural number of processing units.

9. (Original) The method defined in claim 1 wherein the initializing further includes initializing the position of at least part of said machine language routine in the first computing system memory, and the executing includes executing the machine language using the initialized position.

10. (Currently Amended) A machine-readable medium that stores instructions, which when executed by a processor, cause said processor to perform operations comprising:

generating an intermediate language routine from a source language routine provided by a user;

specifying an initial value of each routine variable by calling an initialization routine

~~associated with~~separated from the source language routine and provided by the user, the

initialization routine being called from the intermediate language routine;

performing an optimizing change to the intermediate language routine that results in

an altered intermediate language routine;

generating an object language routine from the altered intermediate language routine;

~~initializing the routine variables using the specified initial values;~~

executing the object language routine in a computer system having architecture of a

target computer system ~~using the initialized values~~;

measuring ~~the~~a characteristic of an execution of the object routine by calling a

measuring routine ~~associated with~~separated from the source language routine and provided by

the user, the measuring routine being called from the intermediate language routine; and

evaluating whether a stopping criterion is met based on a result of calling the

measuring routine and if not,

repeating said performing through said evaluating, and

saving the object language file having a best measured characteristic, until the

stopping criterion is met, wherein the characteristic includes at least one of ~~a timing~~

~~wherein the best measured timing is~~a lowest timing, an object code size, or a bus

utilization,

wherein the initialization routine and the measuring routine are provided by the user as

additional source code separated from the source language routine to be optimized, and

wherein the initialization routine and the measuring routine are compiled, linked, and

launched with the source language routine as a single application.

11. – 12.   (Cancelled)

13.   (Previously Presented) The machine-readable medium in claim 10 wherein the operations further include defining a plurality of segments within the intermediate language routine, each said segment comprising consecutive intermediate code statements wherein no segment includes a same intermediate language routine code, and the performing an optimizing change is performed within one of the segments.

14.   (Previously Presented) The article defined in claim 10 wherein the operations further include determining ordering dependencies in the intermediate language routine, wherein the performing an optimizing change includes maintaining the determined ordering dependencies.

15.   (Previously Presented) The machine-readable medium in claim 10 wherein the optimizing change comprises one of a generic optimization, a reordering, a user selectable reordering, a user selectable global reordering, a user selectable insertion of at least one instruction in a selectable position in the intermediate language routine, and a user selectable removal of at least one instruction from a selectable position in the intermediate language routine; wherein each optimizing change does not affect the intermediate language file integrity.

16.   (Previously Presented) The machine-readable medium in claim 10 wherein the operations further include after the generating the object language routine and before the executing, implementing at least one user selectable optimization to the object language file.

17.   (Previously Presented) The machine-readable medium in claim 10 wherein the

optimizing changes in a sequence of a plural number of a repeated said performing resulting

from the stopping criterion not met is performed according to a process that includes at least

one of a non-repeating optimizing change, a user selectable optimization change sequence;

and a parallel search across a plural number of processing units.

18.   (Previously Presented) The machine-readable medium in claim 10 wherein the

initializing further includes initializing the position of at least a portion of the object language

routine in a main memory of a computing system that performs the execution.

19.   (Previously Presented) The machine-readable medium in claim 10 wherein the

operations further include a user interface for reading from the user at least one of the

specified values of the routine variables, and optimizing instructions wherein the performing

operation includes implementing the optimizing instructions.

20.   (Currently Amended) An apparatus comprising:

        a compiler to translate a source routine into an intermediate routine, and translate an

intermediate routine into an object routine;

        a file initializer to initialize each variable of an object routine to an initial value

according to a user specified value retrieved by calling an initialization routine ~~associated~~

~~with~~ separated from the source routine and provided by a user of the source routine, the

initialization routine being called from the intermediate language routine;

        an optimizer to implement an optimizing change in the intermediate routine;

a computing system having an architecture of a target computing system to execute the optimized object file from a main memory of the computing system in which the variables are assigned the initial values;

an execution characteristic measurer to measure a characteristic of an execution of the object routine including calling a measuring routine ~~associated with~~ separated from the source routine and provided by the user, the measuring routine being called from the intermediate language routine; and

an evaluator to determine if characteristics of the execution of the optimized object routine is in accordance with a stopping criterion based on a result of calling the measuring routine and if not, to repeat an execution of the optimizer, an execution of the re-optimized code, and an execution of the execution characteristic measurer, until the stopping criterion is met, wherein the characteristic includes at least one of a ~~timing wherein the best measured timing is the~~ lowest timing, a machine language size, and a bus utilization,

wherein the initialization routine and the measuring routine are provided by the user as additional source code separated from the source language routine to be optimized, and wherein the initialization routine and the measuring routine are compiled, linked, and launched with the source language routine as a single application.

21.   (Cancelled)

22.   (Original) The apparatus defined in claim 20 wherein the optimizer further includes in a sequence of repeated executions of the optimizer, and execution of the optimized object code, at least one of a non-repeating optimizing change, and a user selectable optimization change sequence.

23. (Original) The apparatus defined in claim 20 wherein the computing system executes an object file that is not optimized in the target computing system from the main memory, that includes the initialized variables and an execution of the execution characteristic measurer.

24. (Cancelled)

25. (Original) The apparatus defined in claim 20 further including a code segmenter to determine a plurality of segments within the intermediate routine, each segment comprising consecutive intermediate routine statements wherein no segment includes a same intermediate language statement, and the optimizer implements a change to one of the segments.

26. (Original) The apparatus defined in claim 20 wherein the optimizing change includes a determination of ordering dependencies in the intermediate file and the optimizing change maintains the determined ordering dependencies.

27. (Original) The apparatus defined in claim 20 wherein the optimizing change includes one of a generic optimization, a reordering, a user selectable reordering, a user selectable global reordering, a user selectable insertion of at least one instruction in a selectable position in the intermediate language routine, and a user selectable removal of at least one instruction from a selectable position in the intermediate language routine; and wherein each optimizing change does not affect the intermediate file integrity.

28.   (Original) The apparatus defined in claim 20 wherein the change in the intermediate routine file includes a user selectable optimization.

29. (Original) The apparatus defined in claim 20 wherein the file initializer is further to initialize the position of at least a portion of the object routine file in the main memory, and the computing system is further to allocate the main memory according to the initialized positions of the portion of the object routine.

30. (Original) The apparatus defined in claim 20 wherein the computing system includes a plurality of processors that each have an architecture of the target computing system.